

Prozedurale Generierung von Baum-Strukturen innerhalb der Unreal Engine 4

David Liebemann

Bachelor-Abschlussarbeit

Betreuer:

Prof. Dr. Christof Rezk-Salama

Trier, 26.02.2017

Kurzfassung

Inhaltsverzeichnis

1	Einleitung und Problemstellung	1
1.1	Prozedurale Generierung	1
1.2	Unreal Engine 4	1
1.3	Ansatz	1
2	Lindenmayer-Systeme	2
2.1	Kontextfreie Grammatik	2
2.2	D0L-Systeme	3
2.2.1	Parametrische L-Systeme	4
2.3	Grafische Interpretation von L-Systemen	4
2.3.1	Turtle-Interpretation	5
2.3.2	Verzweigte L-Systeme	5
2.4	L-Systeme für Baumstrukturen	5
3	Space-Colonization Algorithmus	6
3.1	Ursprung	6
3.2	Erweiterung in dreidimensionalen Raum	6
3.3	Einfluss der Parameter	6
4	Implementierung	7
4.1	Baumstruktur	7
4.2	L-Systeme	7
4.2.1	L-System-Plant	7
4.2.2	Turtle-Graphic-Interpreter	7
4.2.3	Performanz	7
4.3	Space-Colonization-Algorithmus	7
4.3.1	Colonization Space	7
4.3.2	Space-Colonization-Plant	7
4.4	Mesh-Generierung	7
5	Vergleich	8
5.1	Visuell	8
5.2	Performanz	8

6 Zusammenfassung und Ausblick	9
6.1 Erweiterungen	9
6.1.1 Texturen	9
6.1.2 Blätter	9
6.1.3 Generierung zur Laufzeit	9
6.1.4 Verteilung	9
Literaturverzeichnis	10
Glossar	11
Erklärung der Kandidatin / des Kandidaten	12

Abbildungsverzeichnis

2.1	n-fache Ableitung der Anfangszelle a_r anhand der Produktionsregeln $p_1 \dots p_4$	2.1. Eigene Abbildung anhand [PL04, S.4]	4
-----	--	---	---

Einleitung und Problemstellung

1.1 Prozedurale Generierung

Insbesondere Vegetation - warum, was sind die Schwierigkeiten, wenn man es nicht prozedural generiert.

1.2 Unreal Engine 4

Allgemeine Erklärung - was ist die Unreal Engine 4, was stellt sie zur Verfügung, wie entwickelt man dafür? Insbesondere Eingabe von Parametern über den Editor – wichtig für Implementierung.

1.3 Ansatz

Verwendete Ansätze: L-Systeme mit Turtle-Graphik und Space Colonization Algorithmus. Kleine Einleitung zu beiden Ansätzen, bisherige Verwendung.

Lindenmayer-Systeme

Vorstellung der verwendeten Konzepte.

2.1 Kontextfreie Grammatik

Bei L-Systemen handelt es sich um, auf Zeichenketten basierende, Ersetzungssysteme. Es werden komplexe Objekte beschrieben, indem Teile der Zeichenkette durch andere Zeichen oder Zeichenketten ersetzt werden. Die Beschreibung dieser Ersetzungen findet mittels festgelegter Produktionsregeln statt. [PL04, S.2]

Eine formale Definition eines auf Zeichenketten arbeitenden Ersetzungssystems wird durch kontextfreie Grammatiken gegeben:

Kontextfreie Grammatik: 2.1 *Eine kontextfreie Grammatik G ist ein Tupel $G = (V, N, P, \omega)$ bestehend aus:*

1. V – einer nichtleeren, endlichen Menge von Buchstaben (Alphabet).
2. N – einer endlichen Menge von Variablen.
3. P – einer endlichen Menge von Produktionsregeln in der Form $P : A \rightarrow \alpha$ mit $A \in N$ und $\alpha \in (V \cup N)^*$.
4. $\omega \in N$ – dem Axiom, Startsymbol der Grammatik.

[Sch, S.343]

Die Menge V^* ist die Menge aller Wörter über V – die Menge aller Wörter, die aus dem Alphabet V gebildet werden können. [Sch, S.70]

Eine Grammatik wird als kontextfrei bezeichnet, wenn beispielsweise die Produktionsregel $A \rightarrow \alpha$ angewendet werden kann, ohne die A umgebenden Buchstaben – seinen Kontext – beachten zu müssen. [Sch, S.343]

Eine Grammatik wird als deterministisch bezeichnet, wenn es genau eine Produktionsregel $r \in P$ für jede Variable $A \in V$ gibt, sodass $r : A \rightarrow \alpha$, $\alpha \in (V \cup N)^*$. Das bedeutet, dass die Ersetzung einer Variable eindeutig durch eine einzige Regel beschrieben wird. [STN16, S.75]

Die Anwendung der Produktionsregeln findet meist sequentiell statt – die Zeichenkette wird von links nach rechts durchlaufen und Ersetzungen werden direkt auf die untersuchte Zeichenkette angewendet. [STN16, S.75]

2.2 D0L-Systeme

Diese Arbeit beschränkt sich auf die Behandlung deterministischer, kontextfreier L-Systeme, auch D0L-Systeme genannt. Diese besitzen die Eigenschaften einer deterministischen und kontextfreien Grammatik, Produktionsregeln werden jedoch parallel und gleichzeitig auf alle Buchstaben des untersuchten Wortes angewendet. Dieses Vorgehen soll die Zellteilung in mehrzelligen Organismen simulieren und ist somit an biologische Vorgänge angelehnt. [PL04, S. 3]

Ein D0L-System kann wie folgt definiert werden:

D0L-System: 2.1 *Ein D0L-System ist ein Tupel $G = (V, P, \omega)$, bestehend aus*

1. V – einem nichtleeren, endlichen Alphabet.
2. P – einer endlichen Menge von Produktionsregeln in der Form $P : a \rightarrow b$ mit $a \in V$ und $b \in V^*$. a wird als Vorgänger, b als Nachfolger bezeichnet. Ist für einen Buchstaben $x \in V$ keine explizite Produktionsregel angegeben, wird die Regel $P : x \rightarrow x$ angenommen – der Buchstabe wird durch sich selbst ersetzt.
3. $\omega \in V^+$ – dem Axiom, Startsymbol der Grammatik.

Die Menge V^+ ist die Menge aller nichtleeren Wörter über V . [PL04, S.4]

Die Ableitung eines Wortes entspricht der Ersetzung aller Buchstaben anhand der Produktionsregeln. Ein Wort kann mehrmals abgeleitet werden.

Ableitung: 2.1 *Gegeben sei ein Wort $w = a_1 \dots a_m$ mit $w \in V^*$ und $a_i \in V^*$. Das Wort $v = b_1 \dots b_m$ mit $v \in V^*$ und $b_i \in V^*$ ist die Ableitung von w wenn für alle $i = 1 \dots m$ eine Produktionsregel $a_i \rightarrow b_i$ existiert. Die Ableitung wird als $w \Rightarrow v$ notiert.*

Das Wort w_n ist die n -te Ableitung des Wortes w_0 wenn eine Folge von Wörtern w_0, w_1, \dots, w_n mit Ableitungen $w_0 \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_n$ existiert. [PL04, S.4]

Anabaena

Beispiel: Das Wachstum der Blaualgen-Gattung „Anabaena“ kann durch ein L-System simuliert werden. Die Buchstaben a und b beschreiben die Größe und Teilungsbereitschaft einer Algenzelle, während die Indizes l und r die Polarität einer Zelle darstellen. Es gelten folgende Produktionsregeln:

$$\begin{aligned}
 p_1 : a_r &\rightarrow a_l b_r \\
 p_2 : a_l &\rightarrow b_l a_r \\
 p_3 : b_r &\rightarrow a_r \\
 p_4 : b_l &\rightarrow a_l
 \end{aligned} \tag{2.1}$$

Die Entwicklung einer Anfangszelle a_r (Axiom $\omega : a_r$) läuft daraufhin wie in Abbildung 2.1 dargestellt ab.

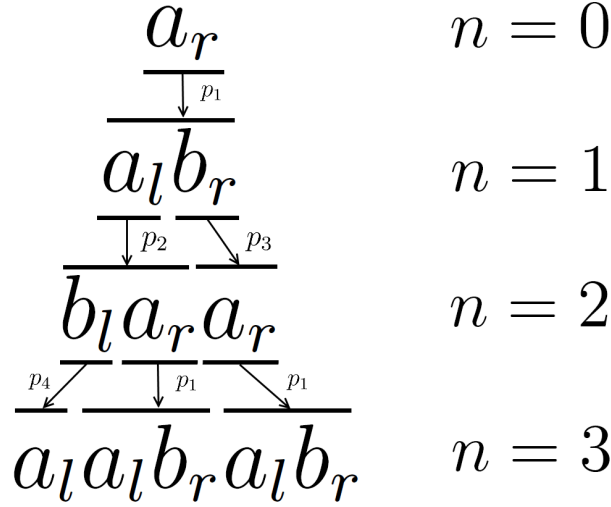


Abb. 2.1. n -fache Ableitung der Anfangszelle a_r anhand der Produktionsregeln $p_1 \dots p_4$ 2.1. Eigene Abbildung anhand [PL04, S.4]

2.2.1 Parametrische L-Systeme

Parametrische L-Systeme stellen eine Erweiterung der D0L-Systeme dar. Die Buchstaben eines verwendeten Alphabets V werden um zugeordnete Parameter aus der Menge der reellen Zahlen ergänzt. Ein solches parametrisches Wort $V \times \mathbb{R}^*$ besteht aus einem Zeichen $A \in V$ und Parametern $a_1, \dots, a_n \in \mathbb{R}$ und wird als $A(a_1, \dots, a_n)$ dargestellt. [PL04, S.41]

Während in der Definition von Modulen numerische Konstanten verwendet werden, werden bei der Angabe eines L-Systems Variablen verwendet. Ist Σ eine Menge von Variablen, dann ist $E(\Sigma)$ ein arithmetischer Ausdruck, in dem Variablen, Konstanten und arithmetische Operatoren auf eine zulässige Weise kombiniert werden. [PL04, S.41]

Parametrisches L-System: 2.2.1 *Ein Parametrisches L-System ist ein Tupel $G = (V, \Sigma, P, \omega)$, bestehend aus*

1. V – einem nichtleeren, endlichen Alphabet.
2. Σ – einer Menge der Variablen.
3. P – einer endlichen Menge von Produktionsregeln $P : (V \times \Sigma^*) \rightarrow (V \times E(\Sigma))^*$
4. $\omega \in M^+$ mit $M = (V \times \mathbb{R}^*)$ – einem Axiom in Form eines nichtleeren, parametrischen Wortes.

[PL04, S.41]

Beispiel:

2.3 Grafische Interpretation von L-Systemen

Um die Ergebnisse von L-Systemen zu visualisieren, muss eine grafische Interpretation der resultierenden Zeichenketten festgelegt werden. Im Folgenden wird die verwendete Visualisierungsmethode, Turtle-Interpretation, vorgestellt.

2.3.1 Turtle-Interpretation

Die Turtle-Interpretation entspricht einer

2.3.2 Verzweigte L-Systeme

asd

2.4 L-Systeme für Baumstrukturen

asd

Space-Colonization Algorithmus

3.1 Ursprung

3.2 Erweiterung in dreidimensionalen Raum

3.3 Einfluss der Parameter

Implementierung

4.1 Baumstruktur

4.2 L-Systeme

4.2.1 L-System-Plant

4.2.2 Turtle-Graphic-Interpreter

4.2.3 Performanz

4.3 Space-Colonization-Algorithmus

4.3.1 Colonization Space

4.3.2 Space-Colonization-Plant

4.4 Mesh-Generierung

Vergleich

5.1 Visuell

5.2 Performanz

Zusammenfassung und Ausblick

Zusammenfassung

6.1 Erweiterungen

6.1.1 Texturen

6.1.2 Blätter

6.1.3 Generierung zur Laufzeit

6.1.4 Verteilung

Literaturverzeichnis

- PL04. PRUSINKIEWICZ, PRZEMYSŁAW und ARISTID LINDENMAYER: *The Algorithmic Beauty of Plants*. Przemyslaw Prusinkiewicz, eBook Auflage, 2004. <http://algorithmicbotany.org/papers/abop/abop.pdf>.
- Sch. SCHMITZ, PROF. DR. HEINZ: *Theoretische Informatik - Vorlessungsskript*.
- STN16. SHAKER, NOOR, JULIAN TOGELIUS und MARK J. NELSON: *Procedural Content Generation in Games*. Springer International Publishing Switzerland 2016, 2016.

A

Glossar

B

Erklärung der Kandidatin / des Kandidaten

☐ Die Arbeit habe ich selbstständig verfasst und keine anderen als die angegebenen Quellen- und Hilfsmittel verwendet.

☐ Die Arbeit wurde als Gruppenarbeit angefertigt. Meine eigene Leistung ist ...

Diesen Teil habe ich selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Namen der Mitverfasser: ...

Datum

Unterschrift der Kandidatin / des Kandidaten